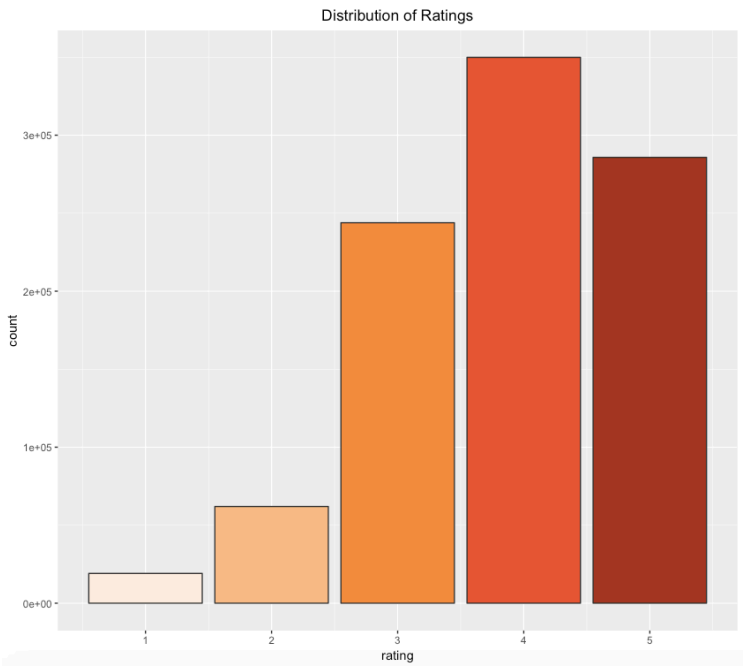
1. Dataset description

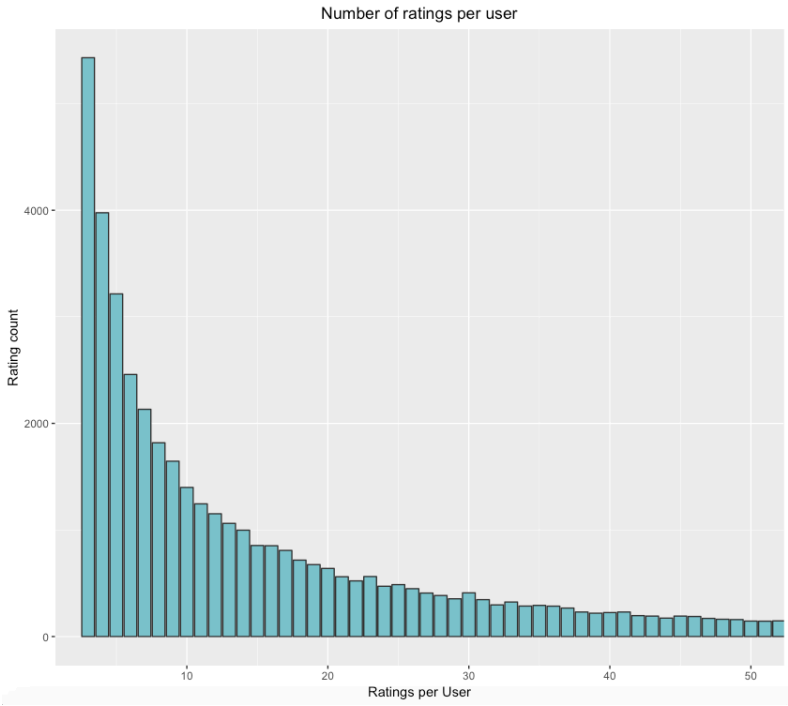
ratings.csv – contains all users' ratings of the books (980k ratings, for 10k books, from 53424 users)
books.csv – contains information on books such as author, year, etc.
book_tags.csv – contains all tag_id's users have assigned to that book and corresponding tag_counts
tags.csv – contains the tag_names corresponding to tag_id's
Last two files are linked by the book_id.

2. Data Exploration

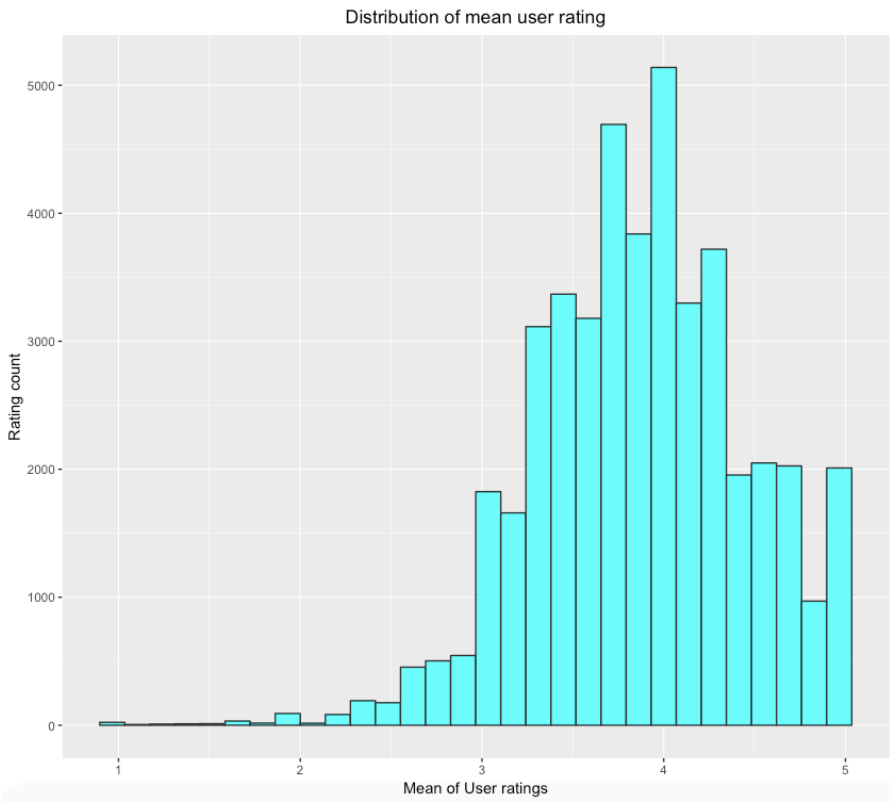
i. *Distribution of ratings:* Most of the ratings are in the 3-5 range, hence, people tend to give ratings from average to excellent.



ii. *Number of ratings per user:*



iii. Distribution of mean user rating:



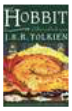
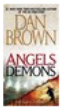
Some people give 5 rating to a mediocre book while others don't unless its excellent. From the plot above, it can be seen that on the right side of the bump are the ones rated with a mean of 5 by the users, which means the users really liked that book.

iv. 10 highly rated books:

	image	title	ratings_count	average_rating
1		The Complete Calvin and Hobbes	28900	4.82
2		Words of Radiance (The Stormlight Archive, #2)	73572	4.77
3		Harry Potter Boxed Set, Books 1-5 (Harry Potter, #1-5)	33220	4.77
4		ESV Study Bible	8953	4.76
5		Mark of the Lion Trilogy	9081	4.76
6		It's a Magical World: A Calvin and Hobbes Collection	22351	4.75
7		Harry Potter Boxset (Harry Potter, #1-7)	190050	4.74
8		There's Treasure Everywhere: A Calvin and Hobbes Collection	16766	4.74
9		Harry Potter Collection (Harry Potter, #1-6)	24618	4.73
10		The Authoritative Calvin and Hobbes: A Calvin and Hobbes Treasury	16087	4.73

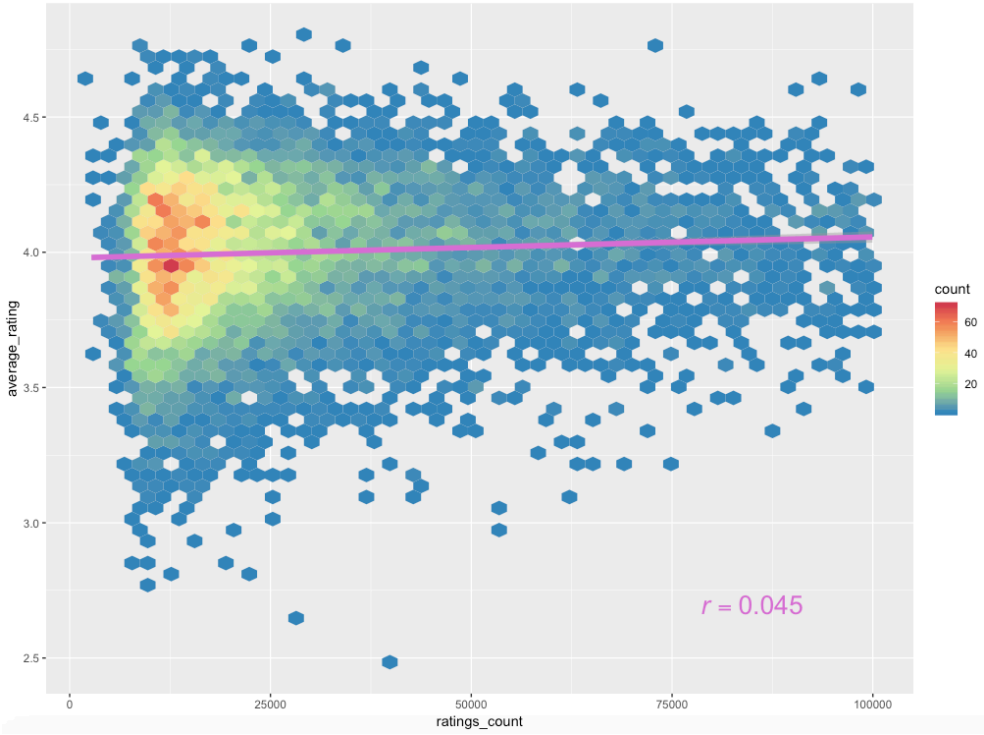
v. 10 most popular books: Books that were rated more often.

	image	title	ratings_count	average_rating
1		The Hunger Games (The Hunger Games, #1)	4780653	4.34
2		Harry Potter and the Sorcerer's Stone (Harry Potter, #1)	4602479	4.44
3		Twilight (Twilight, #1)	3866839	3.57

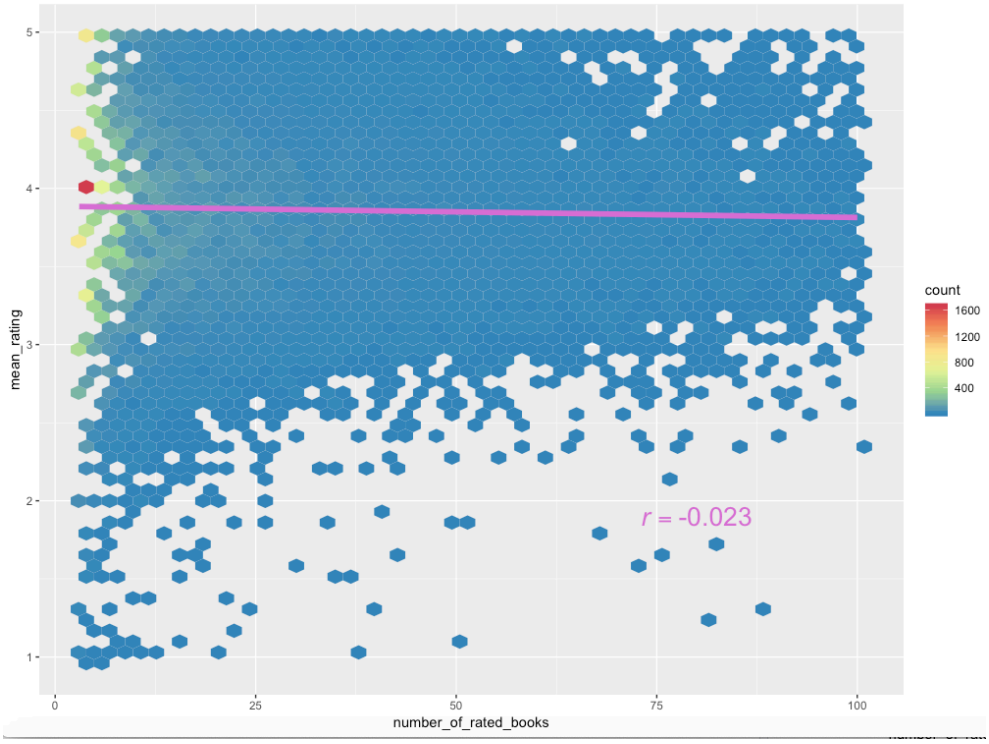
4		To Kill a Mockingbird	3198671	4.25
5		The Great Gatsby	2683664	3.89
6		The Fault in Our Stars	2346404	4.26
7		The Hobbit	2071616	4.25
8		The Catcher in the Rye	2044241	3.79
9		Pride and Prejudice	2035490	4.24
10		Angels & Demons (Robert Langdon, #1)	2001311	3.85

vi. *Factors influencing book's rating:* Features other than this are affecting.

vii. *Relationship between Number of rating and Average rating:* Since value of ‘r’ is very low, there isn’t quite strong relation between the two.



viii. *Frequent raters:* Frequent raters tend to be more critical, hence they don’t give high ratings.



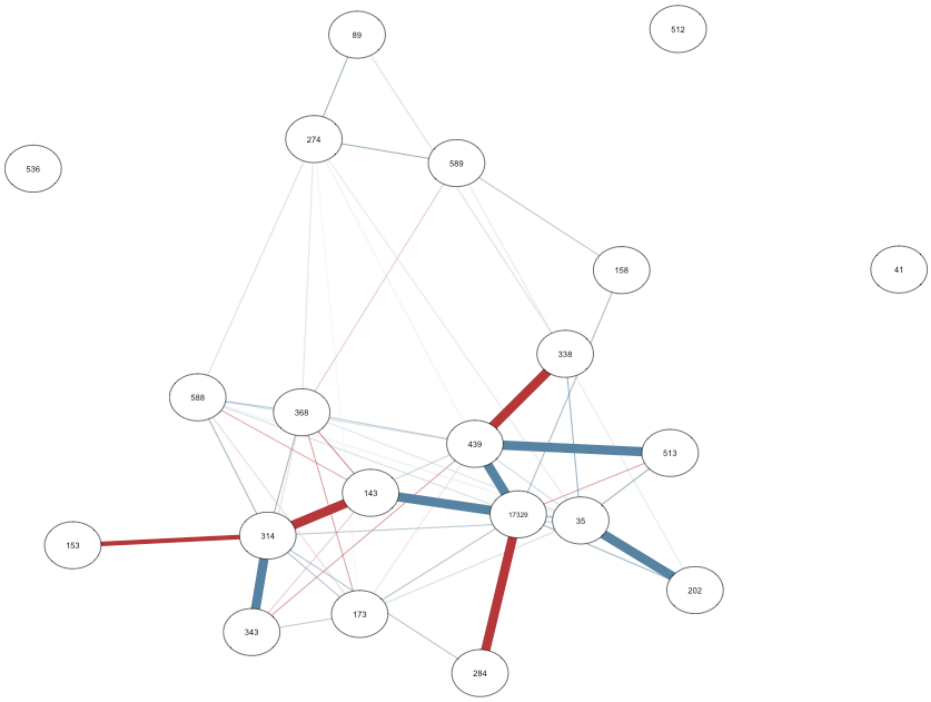
3. Collaborative Filtering

In user-based collaborative filtering, following steps are required to build an algorithm for a recommender system:

1. Identify other users who have similar interests to that of current user in terms of their ratings for the same books.
2. Take the average rating of books the current user has not yet read.
3. Recommend the books with the highest average rating to the current user.

Structuring the data in such a way that each row corresponds to a user and each column to a book.

- Select a user (17329)
- Find similar users
- Normalize user ratings by subtracting the users mean from all individuals.
- Calculate similarity of 17329 with all other users.
- Visualizing the similarities: Blue edges being the most similar and red the least.



- Get predictions for other books: In order to get the recommendations for our user (17329) we would take the most similar users and average their ratings for books our user has not yet rated.

	item	mean_rating
1	1	-0.283236994219653
2	100	0.716763005780347
3	1005	-0.283236994219653
4	1009	-1.48387096774194
5	1017	0.516129032258065
6	102	0.716763005780347
7	1021	-0.424242424242424
8	103	0.716763005780347
9	1063	-1.28323699421965
10	1068	0.716763005780347

- Recommend the best predictions: From above sort the mean ratings and give the best predictions

	mean_rating	book_id	authors	title
1	1.71676300578035	115		
2	1.71676300578035	118		
3	1.71676300578035	1544		
4	1.71676300578035	1597		
5	1.71676300578035	17		
6	1.71676300578035	20		
7	1.71676300578035	27	Bill Bryson	Neither Here nor There: Travels in Europe
8	1.71676300578035	339		
9	1.71676300578035	4		
10	1.71676300578035	520		

- Using Recommenderlab: Recommenderlab is a R-package that provides the infrastructure to evaluate and compare several collaborative-filtering algortihms. Many algorithms are already implemented in the package, and we can use the available ones to save some coding effort, or add custom algorithms and use the infrastructure
- Most of the values in the rating matrix are missing, because every user just rated a few of the 10000 books. Hence, representing this matrix in sparse format in order to save memory.